UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours
Final Paper 2020

Title: Visualizing and Simulating Africa’s Internet Topology

Author: Gerald Joshua Ngumbulu

Project Abbreviation: visAIT

Supervisor(s): Josiah chavula

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	20
Theoretical Analysis	*0*	*25*	
Experiment Design and Execution	*0*	*20*	
System Development and Implementation	*0*	*20*	20
Results, Findings and Conclusions	*10*	*20*	10
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	
Total marks		**80**	**80**

Internet Measurement Platforms used in Discovering Africa's Internet Topology

Gerald Ngumbulu

ngmger002@myuct.ac.za

Department of Computer Science
University of Cape Town

Cape Town, South Africa

ABSTRACT

The question of how efficient the internet measuring techniques used, by internet measurement platforms, to provide a reliable and accurate representation of Africa's internet topology has been raised by many. In this paper, we are going to use three internet measuring platforms to analyze the efficiency of the measuring techniques by analyzing results obtained from each platform. We discover that number of probes in the measurement platforms play a role in a better discovery of Africa's internet topology.

CCS CONCEPTS

• **Networks → Network performance analysis**.

KEYWORDS

Internet measurements, Internet topology, Internet measurement platforms, Simulation

1 INTRODUCTION

The need to better understand how the internet system behaves has increased as internet usage grows in Africa. This has motivated researchers to conduct internet measurements to better understand the current state of the internet. Internet measurements are done so as to understand the internet's dynamic and complex system [12]. However, on a large scale, internet measurements become harder to accomplish. The need to reduce the effort needed in developing and deploying large-scale measurements has resulted in the use of internet measurement platforms. [3]. Internet measurement platforms are distributed sets of dedicated devices that repeatedly run tests on the internet [16]. Platforms implement a range of internet measurement techniques to infer network performance [35].

In this paper, we analyze the efficiency of internet measurement platforms used in collecting data required for the visualization and simulation of Africa's internet topology. The data collected is piped to a web application and used to map Africa's internet for further studies.Section 2 focuses on the background and techniques used by internet measurement platforms. Section 3 examines what has been done in the literature and how past research has informed this research. Section 4 then addresses how we designed and implemented a system that collects data from internet measurement platforms. Section 5 focuses on how we tested the system. Results of the data collected are discussed and analyzed in section 6. Finally, section 7 concludes on what was found from the data collected, how using different measurement platforms impacted the results, how visualization and simulation of internet topology using the data collected fit into this research, and the future works that can be done in this research.

1.1 Problem Statement

The ability to conduct internet measurements via internet measurement platforms has helped in understanding different aspects of the internet and how networks perform in Africa. However, with the current ways of conducting internet measurements, questions on how efficient are the internet measuring techniques used in these platforms arise.

The main research problem we are addressing is how efficient are the internet measuring techniques used by internet measurement platforms to provide a reliable and accurate representation of Africa's internet topology. We solve the main problem by answering questions such as how many internet measurement platforms are we using to analyze how efficient are the current internet measuring techniques used in these platforms. We then look at how we can have a continuous flow of internet measurement results from the chosen platforms to facilitate a continuous analysis of how efficient are the platforms.

1.2 Project objectives

The project objective addressed in this paper is to find out how efficient are the current internet measuring techniques used, by internet measurement platforms, in collecting data for a reliable and accurate representation of Africa's internet topology.

2 BACKGROUND

There are two types of internet measurements, Passive and Active measurements, which are both used to convey information regarding the network infrastructure [26]. This section elaborates on the mentioned types of internet measurements. The section also details the internet measurement platforms that implement a range of measurement techniques to infer network performance in terms of metrics such as latency, packet loss, delays, and throughput [36].

2.1 Passive measurements

These are measurements that are non-intrusive and do not generate additional traffic. Passive measurements observe and collect information that already flows over specific points in the network [43].

Passive measurements uses mechanisms such as RMON and IPFIX [25]. RMON, Remote Monitoring, gathers different types of data such as lost packets and IPFIX gathers IP flow data [42].

The mechanisms explained in the previous sentence collect aggregated data that has little information about the network. Hence, passive measurement techniques such as Simple Network Management Protocol (SNMP) data, NetFlow data and Syslog data are used to collect data produced from the mechanisms such as RMON and IPFIX. Syslog data provides details of failures and activities of networks, NetFlow data provides link utilization information between routers, and Simple Network Management Protocol data provides information such as packet errors at switch and router level [19].

2.2 Active measurements

Active measurements are intrusive measurements that consist of sending probe packets into the network from a source to a destination [41]. The ability to analyze queuing, losses, delays, routing behaviors and propagation delays, are made available by injection of probe packets into the network. The probing should be done politely so as to not affect the data collected.

OWAMP [45], Pathload [38], Pathchar [37], Traceroute and Ping are some of the active measurement tools. The former three use sophisticated packet probing techniques to determine network topology and round-trip delay. Ping and Traceroute are the most commonly used active measurement tools. Unlike the other three, they use ICMP packets to determine network topologies and round-trip delays. ICMP stands for internet control message protocol. ICMP is used to generate error messages when there is a failure in the delivery of the IP packets.

2.2.1 Traceroute

Traceroute is an internet measurement tool that makes it possible to discover paths data packets take from source to destination [27]. Traceroute works by sending multiple ICMP [40] packets with increasing time-to-live fields in the IP header [34]. The host sends an ICMP time exceeded packet to the sender after a packet of time-to-live of one gets discarded when it reaches the host. The time-to-live gets incremented after each respond, which leads to capturing of the IP addresses of the hosts that the packets have traversed heading to the final destination.

2.2.2 Ping

Ping is an internet measurement tool that works by sending an ICMP packet to a destination. The destination sends back an ICMP echo reply packet and the round-trip-time is calculated [18]. Ping is useful in easily finding the IP addresses that can't be reachable from a source. Although Traceroute can also calculate round-trip-time, ping is widely used because it is quicker than Traceroute.

2.3 Internet Measurement Platforms

Internet measurement platforms can be referred to as platforms with probes that repeatedly run network measurement tests on the internet [17]. iPlane [39], Speedchecker [7], Archipelago, RIPE Atlas and DIMES are some of the known internet measurement platforms. In this paper, we are going to test and analyze the efficiency of three platforms: Speedchecker, CAIDA Ark and RIPE Atlas.

2.3.1 RIPE Atlas

RIPE Atlas [46] is an active measurement platform that collects information regarding internet reachability, connectivity, and performance. This information collected can be used to discover internet topology in a specific area. RIPE Atlas is deployed by the RIPE Network Coordination Centre (RIPE NCC) and has hardware probes globally that perform active measurements to collect performance data about the global internet [15].

It has deployed approximately 12k hardware probes around the globe since then. Most of these probes are deployed by network operators in their internal network, and a noticeable number of people volunteer to host a probe at their homes [17].As of 2018 [31], there were 229 active RIPE Atlas probes in Africa alone.

RIPE Atlas is capable of performing ping and traceroute measurements [20]. The measurement results output are in JavaScript Object Notation (JSON) format, which is easy for humans to write and read. Atlas Architecture allows one to run custom measurements, User Defined Measurement (UDM), which consumes credits that can be earned by either sponsoring or hosting a probe [17]. Several researchers have used RIPE Atlas for measurement-based research in Africa (see section 3) and globally.

2.3.2 Archipelago Measurement Platform (CAIDA)

Archipelago [24] is an active measurement platform that is deployed and maintained by CAIDA with two primary goals: support large-scale measurements and collect data to support various research interests. It Currently has 10 active monitors in Africa [6]. Monitors can be referred to as dedicated probes that repeatedly run network measurement tests on the internet. RADclock, Dolphin, and scamper are some of the tools Archipelago uses [24]. Scamper probes the internet to analyze performance and topology [9]. Apart from implementing traceroute, ping, Multi-path Discovery Algorithm (MDA) [13, 14] techniques, scamper also uses Paris-traceroute to control packet header contents and obtain a more precise picture of the specific routes a packet follow [8].

The data from scamper contains meta-data measurement [9]. The data comes out in warts and needs to be converted to JSON format to easily analyze it. Users need an access key to run custom internet measurements.

Archipelago has been used in most of the research studies both in Africa and globally. An example of a measurement hosted by Archipelago is the IPv4 and IPv6 stability that is aimed at comparing performance and reachability. In Africa, most of the research done is aimed at discovering internet topology and measuring network performance (see Section 3).

2.3.3 Speedchecker

Speedchecker is an active measurement platform used for internet research and monitoring Internet infrastructure [7]. It has an API that supports a wide range of network tests from Ping (TCP/ICMP), DNS, Traceroute to HTTP GET. [31].

Apart from being able to handle custom network tests that a user can do via the API, Speedchecker has three types of probes based on the hardware it runs [11]: PC probes, Android probes, and Router probes. They are mostly installed on a windows computer, mobile phones/ tablets, and DD-WRT Routers respectively [11].

Data, such as latency measurements and topology measurements, can be collected from Speedchecker probes and stored in the Speedtest servers [22]. As of 2018 [31], there were 850 probes in Africa covering 52 countries.

3 RELATED WORK

Work done by Formoso et al. [32] presented a measurement campaign methodology that explored the current state of African Internet. They used various vantage points, provided by Speedchecker [10], across the continent to map inter-country delays in Africa. In their paper [32], they showed how data is collected in such a way it can be used in mapping of the internet topology. Most of Speedchecker's probes cover 91% of African countries and are not biased towards university networks. Using Speedchecker has some limitations such as limited insight into the devices launching the measurements. This means that they could not provide causal insight into the performance of individual measurement samples.

Sanby et al. [44] talked about how distributed network probing uses many vantage points to get a more accurate view of network topology. The authors also talked about how varying locations of vantage points further increase the accuracy and completeness of the discovered topology.

Gupta et al.[33] performed traceroutes from access networks to sites hosting popular content. This was done to investigate Internet connectivity in Africa. They increased the number of vantage points, but targeted a small set of African countries [31]. The authors acknowledged that the broadband access networks in the countries that traceroutes were performed from were more developed [23] than in most of the remaining 51 countries [29]. They acknowledged that The result from their research may affect the study as broadband access networks in the countries that traceroutes were performed from do not reflect connectivity in other countries.

The use of distributed network measurement platforms is seen in several research projects done in discovering internet topology and analyzing network performances in Africa. Fanou et al. [28, 29] used data collected in 2014 from RIPE Atlas probes located in African countries to highlight the lack of peering between African ISPs, which result in very high internet traffic delays [28, 29]. Fanou et al. [30] also used RIPE Atlas to analyse the web ecosystem in Africa to shed light on that most of the content accessed by users in Africa is still served from overseas. Chavula et al. [21] researched communications among African research networks. The authors used CAIDA Ark to launch traceroutes to 95 university locations in 29 African Countries. They observed and analyzed how round-trip time is affected and suggested ways to make it better. The measurements lasted for 14 days. Formoso et al [32].

4 REQUIREMENT ANALYSIS AND DESIGN

The primary focus was on designing a tool for collecting measurements from internet measurement platforms. We then tested the designed tool and analyzed the efficiency of the platforms. The model diagrams represented in the section represents the system and not user interaction.

4.1 Requirements Gathering

There was a need to gather requirement so as to ensure that the software system is useful. In our study, we gathered these requirements using few techniques.

4.1.1 Communication with Project Supervisor

The requirements for this research were mostly given by the supervisor. Throughout the project, there was consistent communication with the supervisor via email and almost weekly meetings to track the progress of our project. Halfway through our project, a demo session was scheduled to demonstrate to the supervisor of our project and get feedback.

4.2 Requirements Analysis

Using the above requirement gathering techniques, we managed to gather a list of functional and non-functional requirements. We also managed to show how this system can be integrated into other functions so as to visualize and simulate internet topology in Africa.

After integrating the system into other functions as mentioned in the first paragraph, Users are able to view and analyze the internet topology data collected from three different platforms on a map updated every 3 hours. The three platforms are Speedchecker, RIPEAtlas, and CAIDA Ark. The users see how internet traffic is routed between different autonomous systems and can analyze the RTT (Round-trip-time) of a particular connection between source and destination.

Non-functional requirements that the system met included robustness, Performance, Extensible, data retention and availability.

The system is relatively robust in the sense that it can cope with errors during execution. It is prone to error in data input, since the whole process is automated and designed to accept certain inputs. Anything other than the inputs specified for each script is ignored without breaking the automation process.

It takes 45 minutes for the system to conduct internet measurements, fetch results and create internet topology from the results obtained from each platform. The system performs the automated tasks flawlessly without crashing and can recover if it does crash. The system can also be extended by including extra features such as more internet measurement platforms or by modifying existing functionality for a later upgraded version.

The data in the system is retained until the next batch of internet measurement is done. It is then moved to a different collection to make it possible to see historical trends in the future. This happens every 3 hours. We have documented the system to facilitate ease of use. The functions are also well defined to facilitate the improvement of the code or addition of more features in the future. In terms of availability, the system is available on a web application for anyone who wants to conduct research and study more on how traffic is currently routed in Africa.

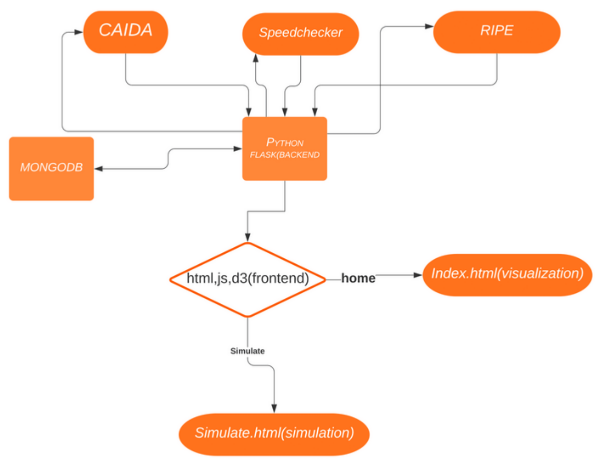

Figure 1: A diagram showing how this part of the research: internet measurement using internet measurement platforms (CAIDA,Speedchecker,RIPE and backend), can be integrated with the rest of the research: Visualizing and simulating Africa's internet topology.

4.3 System Architecture

The architecture of the system can be shown in figure 2. We explained the main components of the system in this section.

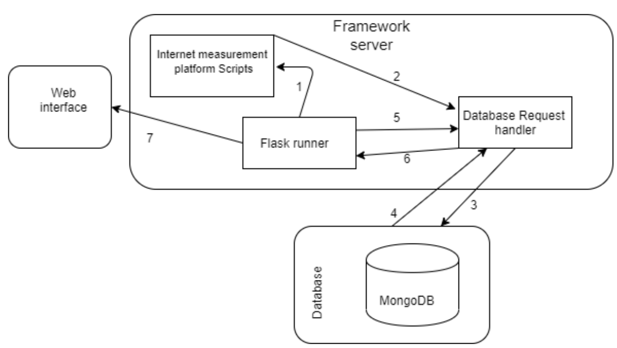

Figure 2: Showing the system overview, and how different components making the system communicate with each other.

The Internet measurement component is responsible for running internet measurement from the three platforms: CAIDA Ark, Speedchecker, and RIPE Atlas. They are responsible for doing traceroute measurements and obtaining results from each platform. Each of the three internet measurement platforms has its own script. The three scripts are ran 8 times a day, everyday. This component uses the list of IP addresses obtained from the IP address Fetcher component (see figure 2).

The Database request handler component analyzes and modifies the data to be stored in the database or passed on for visualization (see figure 2). This component takes the results obtained from the internet measurement component and stores it into a MongoDB database.

IP address fetcher component contains a script that fetches a fresh batch of IP addresses after every 24 hours from all autonomous systems in Africa. The IP addresses are then used as destination for internet measurements. The code justifying how this component works is seen on [5], and the file name is IpFetcher.py

The data storage component consists of a database that is used with the system as seen on figure 2. The database will store the unstructured results returned from the internet measurement platforms. The next step is the analysis and transformation of the data to an internet topology dataset. The result of this analysis is stored in the database where it can be fetched anytime for visualization or simulation purposes.

4.3.1 Storing of Network Measurement Data

Storage of data affects how it is going to be extracted, and how one can run algorithms to analyze the data.

In this system, the data from the storage is transferred to a different collection every three hours and wiped from its previous collection to allow a fresh batch of data to be stored. This is done to display only the current connections when visualizing the data. One of the requirements of the project is that the user needs to get the updated internet topology data every 3 hours. This allows the requirement to be met. However, since we are using a free trial MongoDB online version, we have a limit of how much data we can store. This might result to the loss of historical data when the limit is reached. An upgrade from a free trial to a paid version will solve this issue.

Since some of the data that comes out of the internet measurement platforms are unstructured, data contains nested JSON (values in a JSON object can be another JSON object), the MongoDB database seemed to be a good fit for the storage of the data. From the nature of the project and the requirement explained above, it can be deduced that there would be a lot of reading and writing from the database.

4.3.2 Databases

The system will have one database with nine collections. Each collection stores a different type of unstructured data. After every three hours, the collections are dropped and created to store a new batch of data. However, before the collections are dropped, the data in the collections is transferred to a different collection that stores historical data. There are nine collections, that will be dropped and created, because there are three internet measurement platforms.

Collections for Speedchecker results: these are responsible for storing Speedchecker's trace results, for storing links with RTT(Round trip time) between autonomous systems, and for storing the location of the autonomous systems coming from the GeoLite2 database from maxmind.

Collections for RIPE Atlas results: these are responsible for holding RIPE Atlas' trace results, for storing links with RTT(Round trip time) between autonomous systems, and for storing the location of the autonomous systems coming from the GeoLite2 database from maxmind.

Collections for CAIDA Ark results: these are responsible for holding CAIDA Ark's trace results, for storing links with RTT(Round trip time) between autonomous systems, and for storing the location of the autonomous systems coming from the GeoLite2 database from maxmind.

4.4 Design

4.4.1 Package Diagrams

The models of the system was a result of the high-level system design and requirements. This was used to guide the development of the system. Figure 3 shows the specification of software classes in the system and the associations among them. The system contains 6 classes that represent separate services (see section 4.3 above for explanation).

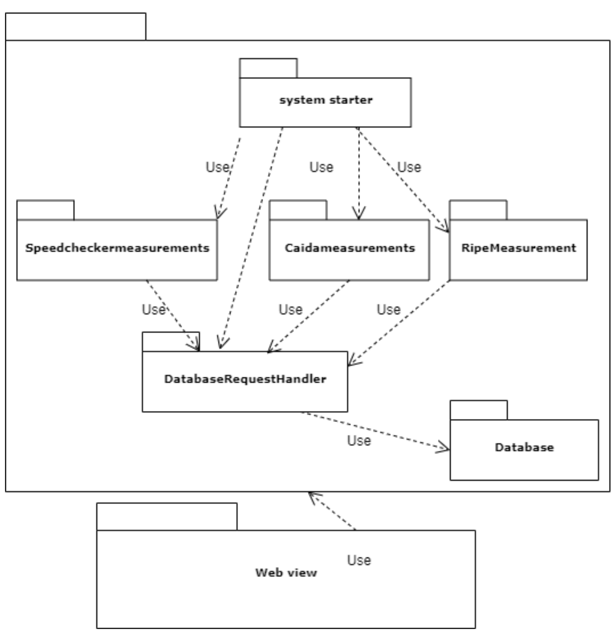

Figure 3: A diagram showing design class diagram of the system.

The system can be integrated on the server and hosted on a web host. This makes the system accessible to other classes such as the web interface, which is responsible for the mapping of the data obtained from our system. There is a use of a database request handler service, in our system to prohibit direct access of the database (see section 4.3 above for explanation).

4.4.2 State and State Transitioning of the System

figure 4 shows the different states of the system. The system has 3 main states: idle state, run measurements state, and fetch IP addresses state. The idle state is the first state that the system is in when it goes live. The system keeps a timer that either triggers a change to run measurement state or fetch IP addresses state. If not, the system remains in the idle state until when it can jump to any of the other two states.

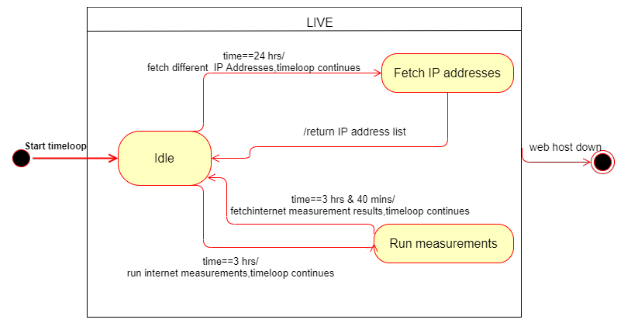

Figure 4: Showing State Machine Chart of the system.

The times at which scripts run prompted this design. Some scripts run after every 24 hours and some run after every 3 hours. The reason we call internet measurement platforms twice, to run measurements and to fetch the results of the measurements done, is because the platforms take time to run measurements to each IP address in our list of IP addresses. This prompts us to wait for sometime before fetching the results of the measurements ran. This makes the system go into the run measurement state twice.

During fetch IP address state, the scheduled script in the system is ran. The result and the system return to the idle state waiting for a jump to the next state.

During run measurements, the script uses the list of IP addresses obtained from the fetch IP address state and run scheduled measurements. They then get results and return to the idle state once it finishes. The idle state jumps into the run measurement state twice. The first time is to run the measurement and the second time is to fetch the results of the measurements done the first time. The system jumps to fetch IP address state and executes a script to get a list of IP addresses after every 24 hours.

The system will forever be in a loop. The system jumps to the run measurement state twice: After every 3 hours to run the measurements from the internet measurement platforms, and 40 minutes after 3 hours to fetch the results of the measurements ran.

5 SYSTEM DEVELOPMENT

5.1 Implementation and Approach

The project required a system that periodically runs internet measurement from internet measurement platforms and stores the data in a database for more analysis. This data is then used in visualization and simulation of Africa's internet topology in the front end web application interface.

Speedchecker [10], RIPE Atlas [4], and CAIDA Ark [3] are the three internet measurement platforms that are used to collect topology data. Topology data collected contains IP addresses of routers encountered from source to destination. The data also contains RTT(Round-trip-time) for the time it took a packet to arrive to its destination and back to its source. Each IP hop is tagged with its geolocation.

Topology data is collected by launching traceroute from all platforms' Africa-based probes to randomly selected destinations located in African countries. The destinations are mostly IP addresses of ASNs in Africa. A script is ran daily to obtain new IP addresses to be used as destinations. The measurements are launched, with a three-hour time-interval, eight times a day. Each probe is configured to launch three consecutive traceroutes to randomly selected IP addresses. For each router hop in the traceroute data, we determine its Autonomous System (AS) using the RIPE Routing Information Service. The location of each router is determined using MaxMind GeoLite2-City [32]. The internet topology dataset formed is restricted to AS level as this is easier to visualize than when it is at the IP level.

Two out of the three measurement platforms, RIPE Atlas and Speedchecker, return data in JSON format. CAIDA Ark returns data in warts. This data is converted to JSON format as well to maintain consistency. There is a test script [2] designed to make sure that correct data is retrieved as expected. The test script checks if data is fetched and compares it with sample data fields expected to see if the fields match.

5.2 Development Framework and Methodology

5.2.1 Using API Documentation from each platform

Each internet measurement platform has a different script that runs to get the intended internet measurement result. The three scripts have similar features.

All the three scripts perform ping and traceroute measurements from source to a specified destination. The Scripts then get results of the ping and traceroute measurement and stores them in an online MongoDB database for further processing. They also obtained the number of active probes in a particular continent or country.

A function that parses the results obtained from CAIDA Ark is added to the CAIDA Ark script to convert data returned to the same format as the other scripts: JSON format.

5.2.2 Programming Language and Framework

The main programming language used to build the system and writing the scripts is Python with flask as a web framework. This made it easy to focus on writing the application and not worry much about the libraries and how the system can be integrated to build a web application. We used git and Github for version control and collaboration. The use of the MongoDB client library for python made it possible to establish a connection with the MongoDB database.

5.3 Software Development Methodology

In this section, we discuss the methodology the system was built on, agile software development approach, and about testing, documentation and maintainability of the system.

5.3.1 Agile Development Methodology

Agile methods are Software Development Methodologies, which center on the idea of iterative development, feedback and change [1]. Thus, the software is built with constant feedback and in a way that is easier for it to incorporate change with changes in requirements. The feedback we got was mostly from the supervisor and our second reader.

5.3.2 Testing, Documentation and Maintainability

We ran the system in an environment which simulated the production environment. This was done so as to test if the non-functional requirements were met (see section 4.2).The system ran internet measurements using the scripts from all internet measurement platforms, fetched the results of the internet measurements and stored them into the MongoDB database. This process was left to run for a week.

We then used the data to plot a cluster column chart showing the maximum round-trip-time value recorded for each destination sources reached from each platform (see section 6.1). We recorded the number of readings obtained from each platform. This helped in analyzing if more probes in a platform led to more routes the platform's traces can discover and result to a better coverage of Africa's internet topology on a map.

We also plotted a separate chart showing the average round-trip-time to the same IP addresses from each of the platforms. This was to test the average time traceroute measurements take to reach to that specific destination from each platform (see figure 12). The number of probes from each platform was recorded to further help in our analysis of whether more probes in a platform resulted to a better coverage of Africa's internet topology on a map (see figure 5).

API testing and unit testing was done so as to make sure each internet measurement platform endpoints return what it was intended to return as explained in the last paragraph of section 5.1. Documentation of the functions is also well done to allow code to be easily be maintained by future developers.

6 RESULTS AND FINDINGS

6.1 Results

This section dives into a detailed discussion of the results from the tests done (see section 5.3.2) on the three platforms.

From Speedchecker, we reached an average of 894 destinations from different sources (probes). We then plotted a chart showing the maximum round-trip-time value recorded for each of the destinations the sources reached (See figure 6).The highest recorded round-trip-time was 4394.33 milliseconds, and the lowest one was below 5 milliseconds (See figure 6). 22.48% of the traces returned an empty ping time array and hence, the round-trip-time for those traces could not be determined.

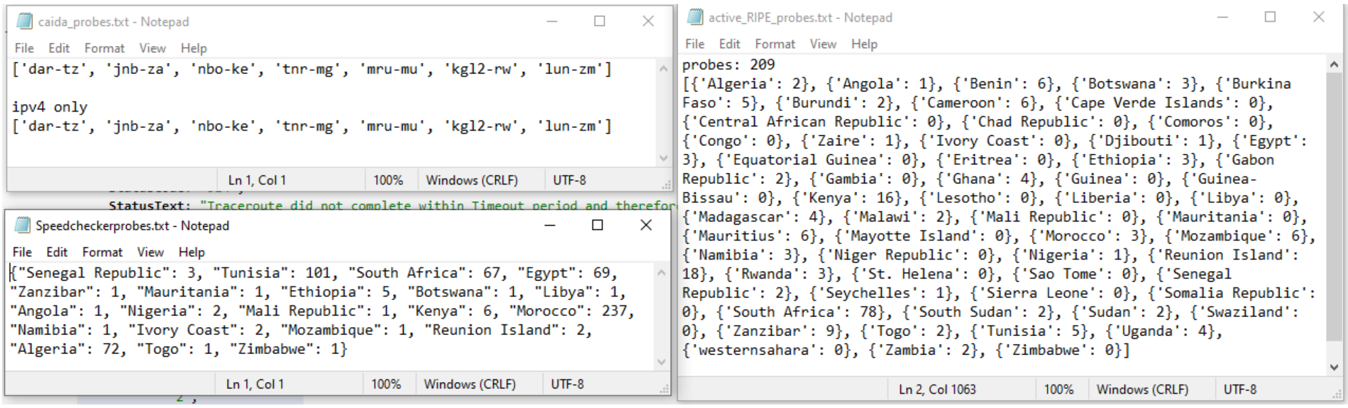

Figure 5: Number of probes from the three platforms

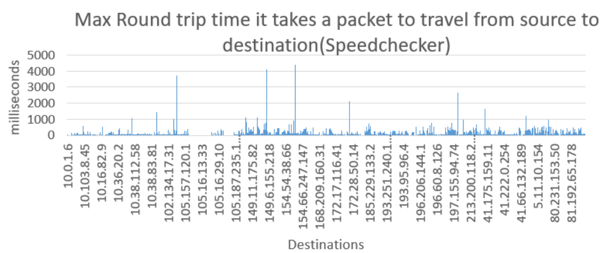

Figure 6: Showing the maximum round-trip time a packet takes to travel from source to destination and back from a Speedchecker platform.

From RIPE Atlas, we reached an average of 333 destinations from different sources (probes). We then plotted a chart showing the maximum round-trip-time value recorded for each of the destinations the sources reached (See figure 7).The highest recorded round-trip-time was 605.33 milliseconds, and the lowest one was below 1 milliseconds (See figure 7). 10.51% of the traces returned an empty ping time array and hence, the round-trip-time for those traces could not be determined.

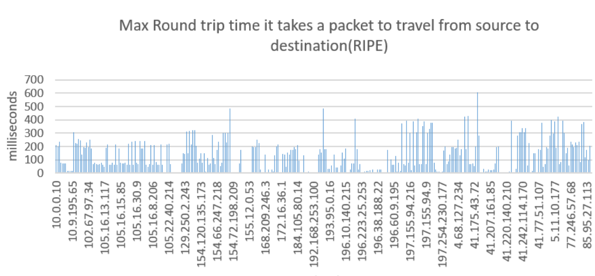

Figure 7: Showing the maximum round-trip time a packet takes to travel from source to destination and back from a RIPE Atlas platform.

From CAIDA Ark, we reached an average of 415 destinations from the different sources (probes). We then plotted a chart showing the maximum round-trip-time value recorded for each of the destinations the sources reached (See figure 8).The highest recorded round-trip-time was 3306.208 milliseconds, and the lowest one was below 1 milliseconds (See figure 8).There were no traces that returned an empty ping time array and hence, all trace's round-trip-time was determined.

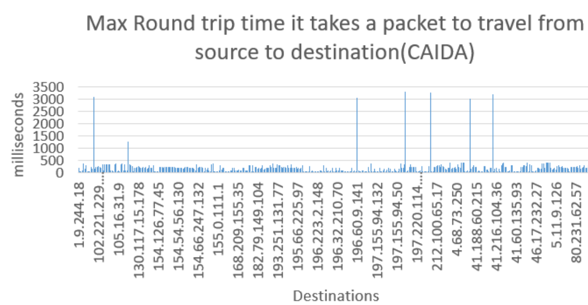

Figure 8: Showing the maximum round-trip time a packet takes to travel from source to destination and back from a CAIDA Ark platform.

After running get probes in Africa, Speedchecker returned an average of 437 available probes. However, 10% of them were unavailable when testing. Out of 54 African countries, an average of 20 had available Speedchecker probes at the time of testing (see Figure 5).

RIPE Atlas returned 1126 probes, 18.56% of the probes were connected and active in Africa (see Figure 5). Most of the active probes where found to be in South Africa as seen on figure 5. CAIDA Ark returned seven available probes. Out of 54 African countries, there were only seven countries with the available probes (see figure 5 for the country names).

As mentioned earlier, we left the system running in an environment that simulated the production environment for three days. The system failed to fetch and update data only when we ran out of credits from RIPE Atlas and when CAIDA Ark stopped every other running measurement to run their own measurement.

However, this did not crash the system or make it unavailable. The system read the previous data it had. We count this unforeseen hiccup in running internet measurements from platforms as another test to see how robust the system is and its ability to prevent itself from crashing and being unavailable. All measurements ran at the same time, and we fixed the fetching of the result 40 minutes after the traceroute measurements were done.

6.2 Findings

From figure 8,6, and 7 we observe that Speedchecker reached more destinations compared to RIPE Atlas and CAIDA Ark. Furthermore, since Speedchecker's probes are more scattered around the continent, it resulted to a better coverage of Africa's internet topology on a map than RIPE Atlas and CAIDA Ark (see figure 9,10, and 11).

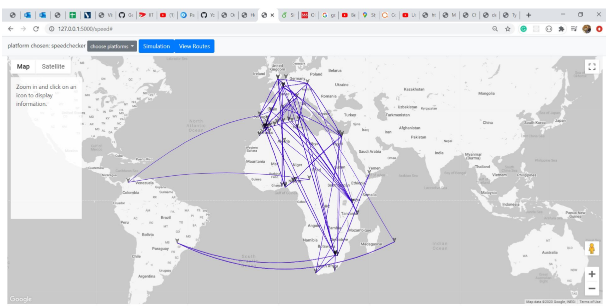

Figure 9: A mapping of Africa's internet topology from Speedchecker platform.

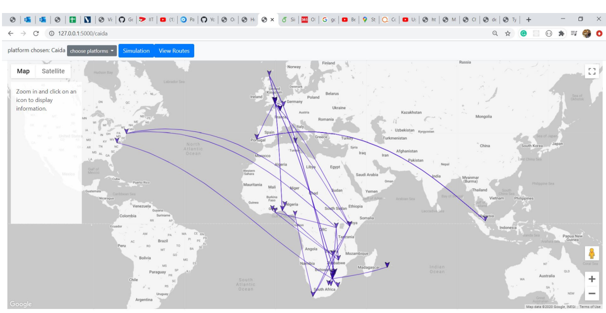

Figure 10: A mapping of Africa's internet topology from CAIDA Ark platform.

from figure 12, we observe that using Speedchecker a relatively lower round-trip-time to a destination was seen. This can also be because of the fact that most of the Speedchecker probes are scattered around compared to RIPE Atlas, and the fact that there are more Speedchecker probes compared to CAIDA Ark (see figure 5).

Approximately half of the RIPE Atlas probes are hosted in South Africa: 78 probes (see Figure 5). Probes from Speedchecker are

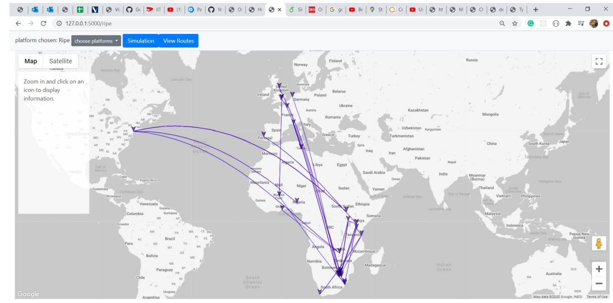

Figure 11: A mapping of Africa's internet topology from RIPE Atlas platform.

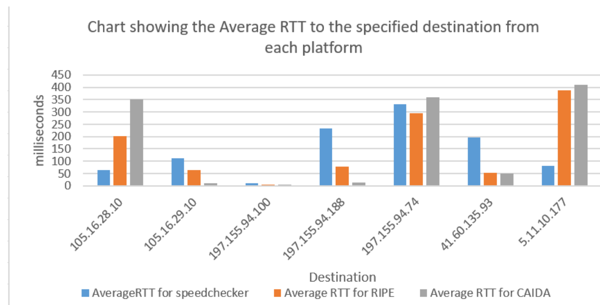

Figure 12: Showing average round-trip time a packet takes to travel from source to the same destination and back from each platform.

more scattered compared to RIPE Atlas. CAIDA Ark has the lowest coverage of probes in Africa (see figure 5). As a result, the northern and western parts of Africa are left without probes. It is worth pointing out that all platforms can be used to collect topology measurements and conduct continuous internet measurements. The ability of the system to continuous run internet measurements is a solution to the issue raised in [36] of how there seemed to be no interest in conducting continuous measurements, which could give a bigger picture of the network.

7 CONCLUSIONS AND FUTURE WORKS

7.1 Conclusion

This work has defined a system to continuously run internet measurements from internet measurement platforms in every 3 hours to discover Africa's internet topology. As mentioned in section 6.2, the system has provided a solution to the issue raised in [36] about not having a continuous measurement of the internet. From the results in section 6.1 we can conclude that more probes in Africa will result to a better coverage of Africa's internet topology on a map. The probes however, need to be scattered all around and not compacted in one area as seen with RIPE Atlas. With the findings seen in section 6.2, All platforms provided accurate representations of Africa's internet topology. However, due to the number of probes and how scattered around the continent they are, Speedchecker discovered more routes from the traceroute measurement done than any of the other two platforms.

7.2 Future Works

This work showed the beginning of using internet measurement platforms to continuously discover Africa's internet topology. The system can be extended to use other internet measurement platforms apart from the three used in this work. The collection and the storing of data in the database can be improved by having another database to store old results for further research purposes.

REFERENCES

[1] Agile methodology. https://www.cprime.com/resources/what-is-agile-what-is-scrum/. (Accessed on 09/15/2020).

[2] Api test file on checking if correct data is returned. https://github.com/Geraldjoshua/Visualization-and-simulation-of-Africa-s-Internet-topology/blob/master/Api_test.py. (Accessed on 09/15/2020).

[3] Caida. https://www.caida.org/research/topology/astopo%20comparisons/. (Accessed on 05/08/2020).

[4] Create measurements · ripe atlas api manual v2. https://atlas.ripe.net/docs/api/v2/manual/measurements/types/. (Accessed on 05/12/2020).

[5] justification on how ip addresses are obtained. https://github.com/Geraldjoshua/Visualization-and-simulation-of-Africa-s-Internet-topology/blob/master/IpFetcher.py. (Accessed on 09/15/2020).

[6] locations. https://www.caida.org/projects/ark/locations/. (Accessed on 09/15/2020).

[7] More on speedchecker. https://www.speedchecker.com/. (Accessed on 09/15/2020).

[8] Paris traceroute. https://paris-traceroute.net/. (Accessed on 09/15/2020).

[9] Scamper active measurement tool. https://www.caida.org/tools/measurement/scamper/. (Accessed on 09/15/2020).

[10] Speedchecker - real world qoe analytics for telecoms and regulators. https://www.speedchecker.com/. (Accessed on 05/29/2020).

[11] Speedchecker probeapi. https://probeapi.speedchecker.com/. (Accessed on 09/15/2020).

[12] Aceto, G., Botta, A., De Donato, W., Marchetta, P., Pescapé, A., and Ventre, G. Open source platforms for internet monitoring and measurement. In *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems* (2012), IEEE, pp. 563–570.

[13] Augustin, B., Friedman, T., and Teixeira, R. Measuring load-balanced paths in the internet. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (2007), pp. 149–160.

[14] Augustin, B., Friedman, T., and Teixeira, R. Measuring multipath routing in the internet. *IEEE/ACM Transactions on Networking 19*, 3 (2010), 830–840.

[15] Bajpai, V., Eravuchira, S. J., Schönwälder, J., Kisteleki, R., and Aben, E. Vantage point selection for ipv6 measurements: Benefits and limitations of ripe atlas tags. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)* (2017), IEEE, pp. 37–44.

[16] Bajpai, V., and Schönwälder, J. A survey on internet performance measurement platforms and related standardization efforts. *IEEE Communications Surveys & Tutorials 17*, 3 (2015), 1313–1341.

[17] Bajpai, V., and Schönwälder, J. A survey on internet performance measurement platforms and related standardization efforts. *IEEE Communications Surveys & Tutorials 17*, 3 (2015), 1313–1341.

[18] Brownlee, N., Loosley, C., et al. 1 fundamentals of internet measurement: A tutorial _.

[19] Calyam, P., Krymskiy, D., Sridharan, M., and Schopis, P. Active and passive measurements on campus, regional and national network backbone paths. In *Proceedings. 14th International Conference on Computer Communications and Networks, 2005. ICCCN 2005.* (2005), IEEE, pp. 537–542.

[20] Canbaz, M. A., Thom, J., and Gunes, M. H. Comparative analysis of internet topology data sets. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (2017), IEEE, pp. 635–640.

[21] Chavula, J., Feamster, N., Bagula, A., and Suleman, H. Quantifying the effects of circuitous routes on the latency of intra-africa internet traffic: a study of research and education networks. In *International Conference on e-Infrastructure and e-Services for Developing Countries* (2014), Springer, pp. 64–73.

[22] Chavula, J., Phokeer, A., Formoso, A., and Feamster, N. Insight into africa's country-level latencies. In *2017 IEEE AFRICON* (2017), IEEE, pp. 938–944.

[23] Chetty, M., Sundaresan, S., Muckaden, S., Feamster, N., and Calandro, E. Measuring broadband performance in south africa. In *Proceedings of the 4th Annual Symposium on Computing for Development* (2013), pp. 1–10.

[24] Claffy, K., Hyun, Y., Keys, K., Fomenkov, M., and Krioukov, D. Internet mapping: From art to science. In *2009 Cybersecurity Applications Technology Conference for Homeland Security* (2009), pp. 205–211.

[25] Claise, B., and Bryant, S. Specification of the ip flow information export (ipfix) protocol for the exchange of ip traffic flow information. Tech. rep., RFC 5101, January, 2008.

[26] Dimitropoulos, X. A., Krioukov, D. V., and Riley, G. F. Revisiting internet as-level topology discovery. In *International Workshop on Passive and Active Network Measurement* (2005), Springer, pp. 177–188.

[27] Donnet, B., and Friedman, T. Internet topology discovery: a survey. *IEEE Communications Surveys & Tutorials 9*, 4 (2007), 56–69.

[28] Fanou, R., Francois, P., and Aben, E. On the diversity of interdomain routing in africa. In *International Conference on Passive and Active Network Measurement* (2015), Springer, pp. 41–54.

[29] Fanou, R., Francois, P., Aben, E., Mwangi, M., Goburdhan, N., and Valera, F. Four years tracking unrevealed topological changes in the african interdomain. *Computer Communications 106* (2017), 117–135.

[30] Fanou, R., Tyson, G., Fernandes, E. L., Francois, P., Valera, F., and Sathiaseelan, A. Exploring and analysing the african web ecosystem. *ACM Transactions on the Web (TWEB) 12*, 4 (2018), 1–26.

[31] Formoso, A., Chavula, J., Phokeer, A., Sathiaseelan, A., and Tyson, G. Deep diving into africa's inter-country latencies. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications* (2018), IEEE, pp. 2231–2239.

[32] Formoso, A., Chavula, J., Phokeer, A., Sathiaseelan, A., and Tyson, G. Deep diving into africa's inter-country latencies. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications* (2018), IEEE, pp. 2231–2239.

[33] Gupta, A., Calder, M., Feamster, N., Chetty, M., Calandro, E., and Katz-Bassett, E. Peering at the internet's frontier: A first look at isp interconnectivity in africa. In *International Conference on Passive and Active Network Measurement* (2014), Springer, pp. 204–213.

[34] Haddadi, H., Rio, M., Iannaccone, G., Moore, A., and Mortier, R. Network topologies: inference, modeling, and generation. *IEEE Communications Surveys & Tutorials 10*, 2 (2008), 48–69.

[35] Isah, M., Phokeer, A., Chavula, J., Elmokashfi, A., and Asrese, A. S. State of internet measurement in africa-a survey. In *International Conference on e-Infrastructure and e-Services for Developing Countries* (2019), Springer, pp. 121–139.

[36] Isah, M., Phokeer, A., Chavula, J., Elmokashfi, A., and Asrese, A. S. State of internet measurement in africa-a survey. In *International Conference on e-Infrastructure and e-Services for Developing Countries* (2019), Springer, pp. 121–139.

[37] Jacobson, V. Pathchar: A tool to infer characteristics of internet paths, 1997.

[38] Jain, M., and Dovrolis, C. Pathload: A measurement tool for end-to-end available bandwidth. In *In Proceedings of Passive and Active Measurements (PAM) Workshop* (2002), Citeseer.

[39] Madhyastha, H. V., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A., and Venkataramani, A. iplane: An information plane for distributed services. In *Proceedings of the 7th symposium on Operating systems design and implementation* (2006), pp. 367–380.

[40] Malkin, G., et al. Traceroute using an ip option. Tech. rep., RFC 1393, January, 1993.

[41] Michaut, F., and Lepage, F. Application-oriented network metrology: Metrics and active measurement tools. *IEEE Communications Surveys & Tutorials 7*, 2 (2005), 2–24.

[42] Mohan, V., Reddy, Y. J., and Kalpana, K. Active and passive network measurements: a survey. *International Journal of Computer Science and Information Technologies 2*, 4 (2011), 1372–1385.

[43] Motamedi, R., Rejaie, R., and Willinger, W. A survey of techniques for internet topology discovery. *IEEE Communications Surveys & Tutorials 17*, 2 (2014), 1044–1065.

[44] Sanby, R., Suleman, H., and Chavula, J. Efficient topology discovery for african nrens. In *2016 IST-Africa Week Conference* (2016), IEEE, pp. 1–8.

[45] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., and Zekauskas, M. A one-way active measurement protocol (owamp). Tech. rep., RFC 4656 (Proposed Standard), 2006.

[46] Staff, R. N. Ripe atlas: A global internet measurement network. *Internet Protocol Journal 18*, 3 (2015).